

대규모 확장이 가능한 범용 신경회로망 : ERNIE

*김 영 주, *정 재 교, **동 성 수, *이 중 호
*인하대학교 정보통신공학과, **용인송담대학 디지털전자정보과
전화 : 032-860-7396 / 핸드폰 : 019-574-2778

Expansible and Reconfigurable Neuro Informatics Engine : ERNIE

*Young Joo Kim, *Jae Kyo Jung, **Sung Soo Dong, *Chong Ho Lee
*Dept. of Information technology & Telecommunication, Inha University
**Dept. of Digital Electronics & Information, Yong-in Songdam College
E-mail : c2021009@inhavision.inha.ac.kr

Abstract

One of the hardest problems in implementation of digital neural network are extension of synapses and programmability for relocating neurons. This paper proposes a new hardware structure to solve these problems. The proposed structure can reconfigure network connections without alteration of basic design, and extend number of synapses attached to one neuron. Also, it is possible to extend the number of neurons and layers by connecting many MPUs(Modular Processing Unit). Generality and extensibility are verified by composing various kinds of Perceptron and Kohonen networks using the architecture proposed in this paper and the verification performances compares well with HDL simulation results as well as the results of C modelling.

I. 서 론

신경회로망은 뇌의 정보처리 방식을 모델화 한 것으로써 복잡한 비선형 시스템의 제어에 적합하며 연상, 추론, 인식 등 기존의 컴퓨터가 해결하기 어려운 응용 분야에서 많이 사용되고 있는 알고리즘이다. 이러한 신경회로망은 병렬, 분산 처리를 본질적인 특징으로 가지기 때문에 소프트웨어 시뮬레이션만으로는 한계를

가지게 되어 1958년 F. Rosenblatt의 MARK I Perceptron 이후로 신경회로망을 하드웨어로 구현하려는 노력이 계속되어져 왔다[1]. 근래에 들어서는 디지털 전자회로기술의 발전에 힘입어 아날로그 신경회로망보다는 디지털 신경회로망의 구현에 더 많은 연구가 있어왔다.

디지털 신경회로망을 구현하기 위해서는 시냅스의 지수적 증가에 따른 면적의 증가, 신경회로망의 재구성, 그리고 네트워크의 확장에 관한 문제점들을 효과적으로 극복할 수 있어야 한다[2]. 따라서 본 논문에서는 SIMD(Single Instruction Multiple Data)구조 및 마스터-슬레이브(Master-Slave)구조를 응용한 새로운 형태의 모듈러 신경망 구조를 제안함으로써 이상의 문제들을 효과적으로 해결하고자 한다.

II. 본 론

2.1 SIMD 구조

하나의 명령으로 여러 유닛들이 동시에 동작하는 SIMD(Single Instruction Multiple Data) 구조는 많은 양의 로컬 데이터를 병렬적으로 처리해야하는 영상 처리나 신경회로망과 같은 어플리케이션을 구현하는데 있어 매우 유용하다. 각각의 PE(Processing Element)는 최소한의 기능만을 수행하는 간단한 구조여야 하며

자신만의 로컬 메모리를 갖고 있다. PE는 특정한 명령을 동시에 받게 되고 자신에게 속해 있는 지역 메모리의 데이터를 해당 명령에 의해 처리하며 필요한 경우 서로간의 연결망을 통해서 처리된 데이터를 주고 받는 것이 가능하다[3]. 이러한 SIMD 구조를 신경회로망에 적용할 경우 PE는 뉴런의 역할을 하게 되고 각각의 PE에 속하는 지역 메모리는 시냅스의 가중치를 저장하는 역할을 하게 된다.

2.2 MPU(Modular Processing Unit)

일반적인 하드웨어 신경회로망에서 뉴런의 시냅스 개수는 PE가 갖고 있는 내부 메모리의 크기에 의해서 결정된다. 따라서 뉴런의 역할을 하는 PE의 개수를 확장한다고 해도 내부 메모리 용량을 초과하는 입력은 받아들일 수가 없기 때문에 뉴런의 개수는 늘어나지만 뉴런 당 입력의 개수는 제한을 받게 된다. 본 논문은 기존의 확장 가능한 신경회로망 구조가 가지고 있는 이러한 결점을 보완하기 위하여 뉴런의 기능을 수행하는 PE를 내부 연결 버스를 통해 묶음으로써 시냅스 개수의 확장이 가능하도록 하는 형태의 하드웨어 구조를 제안하였다. 제안된 구조를 이용하면 시냅스의 확장은 물론 네트워크를 구성하는 가장 큰 기본 모듈인 프로세싱 유닛(Processing Unit)의 연결을 통하여 뉴런 및 레이어의 확장까지 가능하다. SPE(Synapse Processing Element)와 LPE(Layer Processing Element)로 구성되어 있으며 그 자체로서 한 층의 레이어를 구현할 수 있는 이 모듈을 MPU(Modular Processing Unit)라고 명명 하였다.

MPU는 기본적으로 하위 MPU의 출력을 입력으로 받아들이고 자신의 출력은 상위 MPU의 입력으로 사용된다. 따라서 MPU의 입·출력 형식은 동일한 형식을 갖는다. 그림 1은 MPU의 블록 다이어그램이며 이들 PE간에 컨트롤 신호와 데이터가 어떻게 흘러가는지를 보여주고 있다.

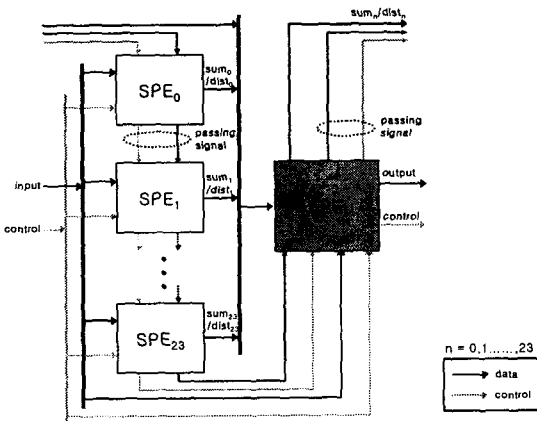


그림 1. MPU의 구성도

제안된 구조는 네트워크의 확장을 위한 두 가지 특성을 갖고 있다. 첫 번째는 MPU간의 확장이며 두 번째는 SPE 간의 확장이다. MPU 간의 확장을 통해서 네트워크를 구성하는 뉴런의 개수 및 레이어의 개수를 확장할 수 있고, SPE 간의 내부 연결 버스를 사용하여 하나의 SPE에서 누적된 시냅스 값을 인접한 SPE로 전달하는 방법으로 시냅스를 확장하는 것이 가능하다. 이와 같은 방법을 통하여 PE가 갖고 있는 지역 메모리의 용량을 초과하는 시냅스를 구현하는 문제를 해결할 수 있다. 이러한 형태의 '시냅스 확장'은 둘 혹은 그 이상의 SPE로 구현하는 것이 가능하며, MPU 내부에서만 국한되는 것이 아니라 외부의 MPU에 속해있는 SPE와도 연결이 가능하기 때문에 대용량의 시냅스를 요구하는 신경망을 매우 융통성 있게 구현할 수 있다. 그림 2는 이와 같은 확장이 어떻게 이루어지는가를 보여주고 있다. 그림 2의 (a)는 두 개의 SPE를 내부 버스로 연결하여 하나의 뉴런으로 구성한 모습이며 (b)는 두 개의 MPU를 연결하여 레이어를 확장한 모습이다.

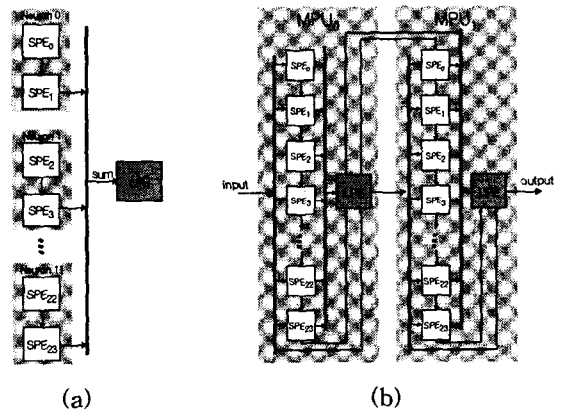


그림 2. 시냅스 확장 및 레이어의 확장 예시

MPU를 구성하는 모든 PE는 동작 초기에 설정 모드를 통하여 그 역할이 결정된다. 이러한 과정을 거침으로써 매우 다양한 형태의 신경회로망을 구성하는 것이 가능하다. 따라서 이러한 하드웨어 구조를 ERNIE (Expsnsible & Reconfigurable Neuro Informatics Engine)라 명명 하였다.

2.3 SPE(Synapse Processing Element)

SPE는 신경회로망의 핵심적인 연산을 담당 하고 있으며 곱셈기, 덧셈기, 누산기의 세 가지 블록으로 구성되어 있다. 퍼셉트론에서 활성화 함수의 입력을 Sum, 코호넨 연산에서 구해지는 입력과 가중치와의 유클리드 거리를 Dist라 하고 x 와 w 가 각각 입력과 가중치

의 n차원 벡터일 때 이들의 관계는 다음과 같이 표현할 수 있다.

$$\begin{cases} Sum = \sum_{i=1}^n x_n w_n \\ Dist = \sqrt{\sum_{i=1}^n (x_n - w_n)^2} \end{cases} \quad (1)$$

그러나 (1)의 유클리드 거리를 구하기 위해서는 실수의 제곱 연산 및 제곱근 연산이 필요하며 이러한 연산들은 하드웨어로 구현하기가 몹시 복잡하다. 그리하여 SPE에서는 (1)대신 (2)와 같은 연산을 수행한다.

$$\begin{cases} Sum_{SPE} = \sum_{i=1}^n x_n w_n + p \\ Dist_{SPE} = \sum_{i=1}^n |x_n - w_n| \end{cases} \quad (2)$$

여기서 p는 시냅스 확장성의 경우 마스터 관계에 있는 SPE로부터 전달 받는 누적합 값을 의미하며 시냅스 확장을 하지 않는 경우나 해당 SPE의 마스터 SPE가 없는 경우에는 p = 0 이 된다. 그림 3은 각각의 경우에 있어서 SPE가 동작하는 모습을 보여준다.

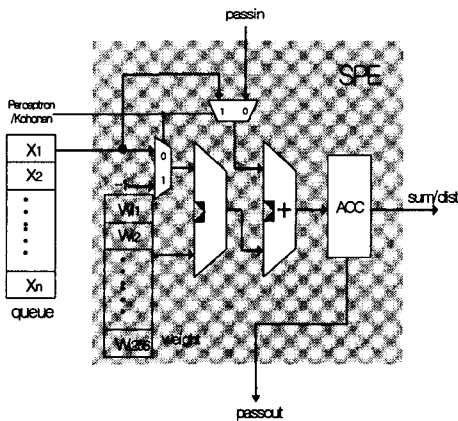


그림 3. SPE의 구성도

2.4. LPE(Layer Processing Element)

LPE는 크게 두 가지 블록으로 나뉜다. 첫 번째 블록은 뉴런의 활성화 함수의 역할을 하는 LUT(Look Up Table)인 PLB(Perceptron Logic Block)이며, 두 번째 블록은 코호넨 층을 구성하는 뉴런 중에서 승자(winner) 뉴런의 인덱스를 출력하기 위한 KLB(Kohonen Logic Block)이다.

III. 실험방법 및 결과

제안된 ERNIE를 Verilog HDL을 이용하여 설계하였다. 설계된 회로의 동작을 검증하기 위하여 신경망을 구성하는데 필요한 데이터를 HDL 검증과 C언어로 작성된 모델링 검증에 동일하게 적용시켰으며 그 실험결과를 서로 비교하였다.

ERNIE의 범용성과 확장성을 증명하기 위하여 표 1과 같이 다양한 종류의 퍼셉트론 및 코호넨 네트워크를 구성하였다. 실험의 편의성을 위하여 퍼셉트론의 경우 SPE 6개와 LPE 1개를 하나의 MPU로 구성하였다. 시뮬레이션 결과와 C 모델링 결과와의 오차는 단층 퍼셉트론의 경우 0.075%였으며 다층 퍼셉트론일 때에는 0.5%로 나타났고 출력층의 활성화 함수로서 계단함수를 사용한 결과 C 모델링 검증 결과와 100% 일치하는 것을 확인할 수 있었다.

표 1. ERNIE를 이용해 구현한 신경회로망의 종류

종 류	비 고	
퍼셉트론	5*6	MPU 1개를 이용하여 신경회로망 구성
	6*8	MPU 2개를 이용하여 신경회로망 구성
	5*8*6	MPU 2개를 이용하여 hidden layer를 구성 MPU 1개를 이용하여 output layer를 구성
	256*32*5	MPU 7개를 이용하여 16*16 픽셀의 인쇄체 문자 인식 application 적용
	300*5	MPU 2개를 이용하면서 SPE 두개로 하나의 뉴런 구성
코호넨 (코호넨층)	8*8	MPU(24SPE/LPE) 3개를 이용하여 8*8의 코호넨층 구성

표 1에서 나타난 바와 같이 간단한 알파벳 인식 실험을 수행하였다. 알파벳 인식 시스템의 신경회로망 구조는 그림 4와 같으며 하나의 알파벳 문자를 픽셀 당 0~255의 값을 가지는 16*16 그레이 스케일의 영상으로 표현한 후 픽셀 값을 -1~1로 정규화 하여 입력으로 사용하였다.

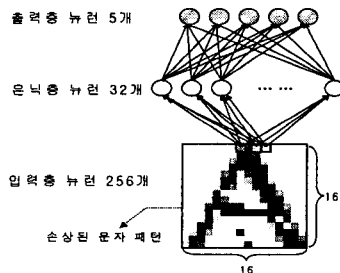


그림 4. 알파벳 인식을 위한 신경회로망 구조

제안된 구조에는 학습 모듈이 내장되어 있지 않기 때문에, PC에서 C언어로 구현된 학습 모델을 통해서 학습된 가중치를 다운로드해서 사용하는 Pre-trained 방법을 사용하였다. 입력 패턴으로는 영어 대문자 26개에 대해서 5가지 폰트를 사용했으며 정상패턴으로 학습을 시킨 후 손상된 패턴을 입력하는 방법으로 인식 능력을 검증 하였다. 이 때 학습율은 0.05이며, 학습에 걸린 반복 횟수는 약 2000회였다. 하드웨어 시뮬레이션은 MPU 7개를 연결하여 256*32*5의 다층 퍼셉트론을 구성한 뒤, 입력 패턴과 가중치를 18bit 데이터 형식으로 변환하여 입력시키는 순서로 실시하였고 이 때 출력된 결과와 C 모델링 결과와의 오차를 확인하였다. 여러 임계 값을 0.001로 주었을 때 HDL 시뮬레이션 결과와 C 모델링의 결과는 알파벳 인식에 있어서 동일한 수준의 인식률을 보여주었다.

또한 우리는 ERNIE의 SOM 구현 능력을 검증하기 위해 T. Kohonen이 실험하였던 내용[4]을 바탕으로 MPU(24 SPE/LPE) 3개를 이용하여 8*8의 코호넨 층을 구성하였다. 입력으로는 표 2와 같이 $x_1 \sim x_5$ 의 속성을 가지는 임의의 패턴을 가정하였으며 초기 학습을 0.04, 최종 학습을 0.008로 하고 8000회 동안 학습시킨 후 얻은 가중치를 ERNIE에 입력하여 검증한 결과 그림 5와 같은 특징 맵이 형성되는 것을 확인할 수 있었다.

표 2. 코호넨 학습에 사용된 패턴 리스트

속성	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13
X1	0.166	0.333	0.5	0.666	0.833	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
X2	0	0	0	0	0	0.166	0.333	0.5	0.666	0.833	0.5	0.5	0.5
X3	0	0	0	0	0	0	0	0	0	0	0.166	0.333	0.5
X4	0	0	0	0	0	0	0	0	0	0	0	0	0
X5	0	0	0	0	0	0	0	0	0	0	0	0	0

	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26
X1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
X2	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
X3	0.666	0.833	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
X4	0	0	0	0.166	0.333	0.5	0.833	1	0.5	0.5	0.5	0.5	0.5
X5	0	0	0	0	0	0	0	0	0.166	0.333	0.5	0.666	0.833

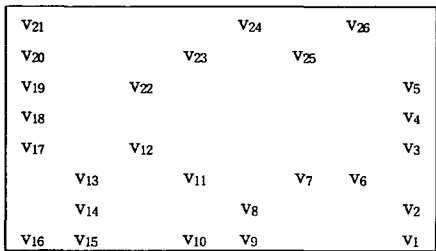


그림 5. 학습된 가중치를 적용했을 때의 특징 맵

IV. 결 론

본 논문에서는 디지털 신경회로망을 구현함에 있어서 발생하는 면적 증가 문제 및 네트워크의 확장을 비롯한 재구성 문제를 효과적으로 개선하기 위해서 제안된 ERNIE를 이용하여 여러 가지 실험을 수행 하였다.

ERNIE는 모듈러 구조를 통하여 얼마든지 확장이 가능하고 다양한 종류의 퍼셉트론 및 코호넨 네트워크를 구성할 수가 있다. 또한 SIMD 구조를 적용하여 하드웨어의 병렬성을 최대로 높였으며 모든 연산을 파이프라인의 형태로 수행하게 함으로써 연산 속도의 측면에서도 그 성능을 극대화 시켰다. 이와 같은 구조적 특성으로 인하여 ERNIE는 다목적 신경회로망 하드웨어 구조로서 다양한 분야에서의 응용이 가능하며 동작 중에 재구성이 가능한 특징을 이용하면 실시간 재구성이 필요한 진화 하드웨어 분야에서 그 효용성이 매우 클 것이라 기대된다.

ERNIE에는 퍼셉트론 및 코호넨 네트워크의 학습을 위한 기능이 구현되어 있지 않으며, 이를 가능케 하기 위해서는 별도의 학습 모듈이 필요하다. 이러한 학습 모듈과 진화를 위한 부가적인 기능이 보장된다면 ERNIE는 많은 공학적 분야에서 더욱 유용하게 사용될 수 있을 것이다.

참고문헌

- [1] Robert J. Schalkoff, *Artificial neural networks*, McGraw-Hill, pp.1-2, pp.411, 1997
- [2] K. Mathia and J. Clark, "On Neural Network Hardware and Programming Paradigms", *Neural Networks*, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on , Volume: 3 , 2002, pp. 2692 -2697
- [3] S. Vitabile, A. Gentile, G.B. Dammone, F. Sorbello, "Multi-layer perceptron mapping on a SIMD architecture", *Neural Networks for Signal Processing*, 2002, Proceedings of the 2002 12th IEEE Workshop on, 2002, pp. 667-675
- [4] T. Kohonen, *Proceedings of the IEEE*, Volume: 78 Issue: 9, pp. 1464 - 1480, Sep 1990